

Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application.

1. (Currently Amended) A system for communications, comprising:

a transport layer/network layer processing stack;

an intermediate driver coupled to the transport layer/network layer processing stack via a first miniport and a second miniport, the intermediate driver concurrently supporting at least two software objects, wherein the first miniport supports a first VLAN group and comprises a first software object that is dedicated to teamed traffic, wherein the second miniport supports a second VLAN group and comprises a second software object that is dedicated to offloaded traffic in a system that can offload traffic, wherein the offloaded traffic in the system that can offload traffic and the teamed traffic pass through the intermediate driver, wherein offloaded traffic in the system that can offload traffic bypasses the intermediate driver;

a first network interface card coupled to the intermediate driver via a first respective physical port, wherein the teamed traffic passes through the first network interface card; and

a second network interface card coupled to the intermediate driver via a second respective physical port, wherein the second network interface card, that is part of the system that can offload traffic, communicates with the first miniport and the second miniport via the second respective physical port, wherein the teamed traffic, the offloaded traffic and the offloaded traffic pass through the second network interface card,

wherein the second network interface card concurrently participates in a team with the first network interface card for a first type of traffic via the first software object that supports the first VLAN group and in ~~[[a]]~~ the system that offloads can offload traffic for a second type of traffic that bypasses the intermediate driver, wherein the system that can offload traffic comprises an upload path that passes through the intermediate driver and an offload path that bypasses the intermediate driver, wherein the second network interface card participates on the

upload path through the second software object that is dedicated to uploaded traffic in the system that can offload traffic,

wherein a third network interface card is coupled to the intermediate driver via a third respective physical port and participates in the team with the first network interface card and the second network interface card via the first software object, and wherein, if [[a]] the third network interface card coupled to the intermediate driver fails, then the first network interface card and the second network interface card handle, via the first software object, a load previously supported by the failed third network interface card;

~~wherein the first miniport supports teaming over the first network interface card and the second network interface card,~~

~~wherein the second miniport provides a dedicated upload path for the second network interface card of the system that can offload traffic from the transport layer/network layer processing stack;~~

~~wherein the second network interface card uses the second miniport for first traffic uploaded from the system that can offload traffic from the transport layer/network layer processing stack and uses the first miniport for second traffic related to the teaming with the first network interface card, and~~

~~wherein the first network interface card uses the first miniport for third traffic related to the teaming.~~

2. (Previously Presented) The system according to claim 1, wherein a first NDIS miniport is communicatively disposed between the intermediate driver and the first interface card, and wherein a second NDIS miniport is communicatively disposed between the intermediate driver and the second network interface card, and wherein a virtual bus driver is communicatively disposed between the second NDIS miniport and the second network interface card.

3. (Previously Presented) The system according to claim 1, wherein the first network interface card comprises a plurality of network interface cards.

4. (Previously Presented) The system according to claim 1, wherein the second network interface card comprises a remote-direct-memory-access-enabled (RDMA-enabled) network interface card.

5. (Previously Presented) The system according to claim 1, wherein the second network interface card is the only network interface card that supports traffic from the system that can offload from the transport layer/network layer processing stack.

6. (Original) The system according to claim 1, wherein the transport layer/network layer processing stack comprises a transmission control protocol/internet protocol (TCP/IP) stack.

7. (Original) The system according to claim 1, wherein the first miniport comprises a virtual miniport instance.

8. (Original) The system according to claim 7, wherein the virtual miniport instance comprises a virtual miniport instance adapted for teamed traffic.

9. (Original) The system according to claim 1, wherein the second miniport comprises a virtual miniport instance.

10. (Original) The system according to claim 9, wherein the virtual miniport instance comprises an RDMA-enabled virtual miniport instance.

11. (Original) The system according to claim 1, wherein the system that can offload traffic from the transport layer/network layer processing stack comprises a Winsock Direct system.

12. (Original) The system according to claim 1, wherein the second miniport supports traffic that is processed by the transport layer/network layer processing stack.

13. (Original) The system according to claim 1, wherein the second miniport supports traffic that has not been offloaded by the system that can offload traffic from the transport layer/network layer processing stack.

14. (Original) The system according to the claim 1, wherein traffic that has been offloaded by the system that can offload traffic from the transport layer/network layer processing stack bypasses the transport layer/network layer processing stack and the intermediate driver.

15. (Previously Presented) The system according to claim 1, wherein the intermediate driver concurrently supports teaming through a first path to the transport layer/network layer processing stack and uploading through a second path to the transport layer/network layer processing stack, the second path being a non-offload path.

16. (Original) The system according to claim 1, wherein the intermediate driver comprises a network driver interface specification (NDIS) intermediate driver.

17. (Original) The system according to claim 1, wherein the intermediate driver is aware of the system that can offload traffic from the transport protocol/network protocol processing

stack.

18. (Original) The system according to claim 1, wherein teaming supports load balancing.

19. (Original) The system according to claim 1, wherein teaming supports fail over.

20. (Original) The system according to claim 1, wherein teaming supports virtual network capabilities.

21. (Currently Amended) A system for communications, comprising:

a first set of network interface cards comprising a second set and a third set of network interface cards, the second set comprising a network interface card that is capable of offloading one or more connections, the third set comprising one or more network interface cards that are not capable of providing an offload path;

an intermediate driver coupled to the second set and to the third set, the intermediate driver being part of a host computer and supporting teaming over the second set and the third set, the intermediate driver concurrently supporting at a first software object and a second software object, wherein the first software object supports a first VLAN group and is dedicated to teamed traffic, wherein the second software object supports a second VLAN group and is dedicated to uploaded traffic in a system that can offload traffic, wherein the uploaded traffic in the system that can offload traffic and the teamed traffic pass through the intermediate driver, wherein offloaded traffic in the system that can offload traffic bypasses the intermediate driver;

a host protocol processing stack coupled to the intermediate driver via the first software object and the second software object ~~via a first virtual miniport instance and a second virtual miniport instance,~~

wherein each network interface card of the second set is coupled to the intermediate driver via a first respective physical port, wherein the teamed traffic and the uploaded traffic pass through the first respective physical port, wherein the teamed traffic passes through the first software object and the first respective physical port, wherein the uploaded traffic passes through the second software object and the first respective physical port, wherein each network interface card of the second set communicates with the first software object and the second software object via the first respective physical port, wherein each network interface card of the second set supports the teamed traffic, the uploaded traffic and the offloaded traffic.

wherein the ~~third~~ second set of network interface cards concurrently participates in a team with the ~~second~~ third set of network interface cards for a first type of traffic and in ~~an~~ offload the system that can offload traffic for a second type of traffic that bypasses the intermediate driver, wherein the first type of traffic is not capable of being offloaded, wherein the second type of traffic is capable of being offloaded.

wherein a fourth set of network interface cards are coupled to the intermediate driver, and wherein, if a particular network interface card in [[a]] the fourth set of network interface cards coupled to the intermediate driver fails, then a plurality of network interface cards from the first set handle, via the first software object, a load previously supported by the failed network interface card of the fourth set;

~~wherein the teamed traffic of the second set and the third set passes through the first virtual miniport instance, and~~

~~wherein uploaded traffic from the offload system passes through only the second virtual miniport instance that is dedicated to the third set, and~~

~~wherein the intermediate driver provides load balancing over some or all of the first set.~~

22. (Previously Presented) The system according to claim 21, wherein the second set provides a kernel bypass path and wherein the third set does not provide a kernel bypass path.

23. (Previously Presented) The system according to claim 21,
wherein the second set is associated with a system that is capable of offloading one or more connections,
wherein the system that is capable of offloading one or more connections offloads a particular connection, and
wherein packets carried by the particular offloaded connection bypass the intermediate driver.

24. (Previously Presented) The system according to claim 21, wherein the intermediate driver provides fail over procedures.

25. (Previously Presented) The system according to claim 21, wherein the host computer communicates, via a team of network interface cards from the second set and the third set, with a remote peer over a network.

26. (Currently Amended) A method for communicating, comprising:
(a) teaming a plurality of network interface cards using an intermediate driver of a host computer, the intermediate driver providing load balancing over some or all of the network interface cards and providing fail over procedures, ~~wherein the teaming is performed by the host computer and/or the plurality of network interface cards~~, wherein the plurality of network interface cards support remote direct memory access (RDMA) traffic, wherein teamed traffic passes through a first miniport that is communicatively disposed between a host TCP/IP stack and the intermediate driver, the intermediate driver concurrently supporting at least two software objects, wherein the first miniport supports a first VLAN group and comprises a first software object that is dedicated to the teamed traffic, wherein the second miniport supports a second

VLAN group and comprises a second software object that is dedicated to uploaded traffic in a system that can offload traffic, wherein the uploaded traffic in the system that can offload traffic and the teamed traffic pass through the intermediate driver, wherein offloaded traffic in the system that can offload traffic bypasses the intermediate driver;

(b) adapting a first network interface card of the plurality of network interface cards to concurrently support at least three paths to an application layer including an offload path, an upload path and a team path, the offload path and the upload path being used for an offload system that can offload traffic for a first type of traffic, the team path being used for a second type of traffic, the upload path passing through a second software object ~~miniport~~ dedicated to the adapted at least one network interface, the second software object ~~miniport~~ being communicatively disposed between the host TCP/IP stack and the intermediate driver, the offload path bypassing the intermediate driver and the host TCP/IP stack, wherein the intermediate driver concurrently supports teaming through the team path and the first software object to the host TCP/IP stack and uploading through the upload path and the second software object to the host TCP/IP stack, wherein the first network interface card is coupled to the intermediate driver through a respective physical port, wherein the teamed traffic passes through the respective physical port and the first software object, wherein the uploaded traffic passes through the respective physical port and the second software object, wherein the offloaded traffic bypasses the intermediate driver and does not pass through the respective physical port, the first software object or the second software object;

(c) adapting remaining network interface cards of the plurality of network interface cards not to provide an offload path, wherein the teamed traffic over the adapted at least one network interface card and the adapted remaining network interface cards passing through the first software object ~~miniport~~; and

(d) if a second network interface card coupled to the intermediate driver fails, then at least two of the plurality of network interface cards handle, via the first software object, a load

previously supported by the failed second network interface card.

27. (Previously Presented) The method according to claim 26, wherein (b) comprises solely associating the offload system that is capable of offloading one or more connections with a single network interface card of the plurality of network interface cards.

28. (Currently Amended) A method for communicating, comprising:
teaming a plurality of network interface cards of a host computer, the plurality of network interface cards not providing an offload path that bypasses a kernel of the host computer;
adding a first additional network interface card to the host computer, the first additional network interface card concurrently supporting at least three paths including an offload path, an upload path and a team path, the offload path of ~~an offload~~ a system that can offload bypassing the kernel of the host computer, the upload path of the ~~offload~~ system that can offload passing through the kernel of the host computer, the ~~offload~~ system that can offload being used for a first type of traffic, the team path being used for a second type of traffic, the first type of traffic being capable of being offloaded by the system that can offload, the second type of traffic not being capable of offloaded by the system that can offload, the upload path passing through a first miniport that is dedicated to uploaded traffic ~~and the first additional network interface card~~, the first miniport being communicatively disposed between an intermediate driver and the host TCP/IP processing stack;
teaming the plurality of network interface cards and the first additional network interface card, the teamed traffic passing through a second miniport communicatively disposed between the intermediate driver and the host TCP/IP processing stack, wherein the first additional network interface card concurrently supports teaming, offloading and uploading;
providing, by the intermediate driver, load balancing over the plurality of network interface cards and the additional network interface card, the intermediate driver being

communicatively disposed between (1) the host TCP/IP processing stack and (2) the plurality of network interface cards and the additional network interface card, wherein the intermediate driver concurrently supports teaming through the team path to the host TCP/IP processing stack and uploading through the upload path to the host TCP/IP processing stack, the intermediate driver concurrently supporting at least two software objects, wherein the second miniport supports a first VLAN group and comprises a first software object that is dedicated to the teamed traffic, wherein the first miniport supports a second VLAN group and comprises a second software object that is dedicated to uploaded traffic in the system that can offload traffic, wherein the uploaded traffic in the system that can offload traffic and the teamed traffic pass through the intermediate driver, wherein offloaded traffic in the system that can offload traffic bypasses the intermediate driver, wherein the first additional interface card is coupled to the intermediate driver through a respective physical port, wherein the teamed traffic passes through the respective physical port and the first software object, wherein the uploaded traffic passes through the respective physical port and the second software object, wherein the offloaded traffic bypasses the intermediate driver and does not pass through the respective physical port, the first software object or the second software object; and

adding a second additional network interface card to the host computer, wherein, if the second network interface card fails, then at least two network interface cards of the first additional network interface card and/or the plurality of network interface cards handle, via the first software object, a load previously supported by the failed second additional network interface card.

29. (Previously Presented) The method according to claim 28, further comprising:

handling packets of a particular connection only via the additional network interface card, the particular connection being maintained by the offload system that is capable of offloading traffic from the host TCP/IP processing stack.

30. (Previously Presented) The method according to claim 28, wherein the intermediate driver provides fail over procedures.

31. (Previously Presented) The method according to claim 28, further comprising:
processing packets of a particular connection via the host TCP/IP processing stack, the particular connection not being an offloaded connection although being maintained by the offload system that is capable of offloading traffic from the host protocol stack.

32. (Original) The method according to claim 31, further comprising:
transmitting the processed packets only through the additional network interface card.